MTBrush_paper

Yixing Tu

3/25/2022

Abstract

MTBrush is a R package that can be used to conduct and visualizing multiple hypothesis tests for high dimensional datasets. It allows users to perform large scale hypothesis testing by fiting a model based on their choices and visualize histograms of test statistics. It also contains an interactive platform allowing users to brush through histograms and query for interesting samples. By comparing the highlighted part in the histograms of different terms and with the help of corresponding scatter plots, users can easily identify whether the interaction terms have synergistic or competitive effects and then make informative conclusions.

Keywords

Multiple hypothesis testing, interaction diagrams, R package;

Introduction

When dealing with high volume dataset like microbiome and microarray study, it is normal to perform large-scale testing and get a huge number of test statistics. It is tedious to analyze results from the multiple hypothesis testing procedure due to the large amount and the some seemed significant results due to the tolerant significant level. Visual graphs for statistics can help to grasp some information from the data. Studies involved multiple hypothesis testing use histogram of test statistics for each sample to help making interpretations, but this can be still hard to interpret. They are lacking the part of linking test statistics to the information form original dataset. It is especially hard to interpret plots as plots getting complex in some high dimensional and multivariate studies (Wills, 08). The solution is using linked data views. Not trying to make a complex plots explain everything from the data and test results, we could build several simpler parts and link them together to help to analyze the data. Brushing has been used for connected statistics on multiple levels to one another (Wickham), but there isn't one for models under multiple hypothesis settings. Therefore, this package is trying build the linked visualizations for multiple hypothesis testing studies.

idea: replace the traditional way of calculating log fold-changes, instead we fit ANOVA models.

Methods

Implementation

The package contains five methods:

split_dataset(df, group): This function splits the given data set into many smaller subsets based on the group. Each subset only contains measurements of a sample and all subsets take same measurements.

parameters:

df: the given data frame;
group: the name of the group column

return: groups of subsets

fit_statistics(subsets, lm_func, group): This method helps to fit a model chosen by users for all groups of subset.

parameters:

subsets: groups of subset from split_dataset function;
lm_func: the linear model chosen by user to fit the data;
group: the name of the group column

return: a table with statistics of each term for all subsets

draw_stats_histogram(stats_df): This function provides a part of the shiny app to generate several histograms of statistics for each term

parameters:

stats_df: the output table from fit_statistics function

return: several histograms of all subsets based on terms in the model

brush_plots_binary(df, stats_df, group_list, group, value): This method generates a shiny app displaying histograms of statistics and scatter plots and a table of selected observations. Use this function when the condition/explanatory variables have binary data type.

parameters:

df: the processed data frame;
stats_df: the output table from fit_statistics function;
group_list: a list of distinct observations;
group: the column name for grouping variable;
value: the column name for response variable;

return: the shiny user interface contains histograms of statistics and scatter plots and a table of selected observations.

brush_plots_other(df, stats_df, group_list, group, value): This method is similar to brush_plots_binary. It also generates a shiny app displaying the histograms of statistics and scatter plots and a table of selected observations. Use this function when the condition/explanatory variables have non-binary data type.

parameters:

df: the processed data frame;
stats_df: the output table from fit_statistics function;
group_list: a list of distinct observations;
group: the column name for grouping variable;
value: the column name for response variable;

return: the shiny user interface contains histograms of statistics and scatter plots and a table of selected observations.

Operation

Before users make any interactions, the user interface of the generated Shiny app is made of a group of histograms of statistics and a scatter plot for all samples. Users can brush through any histogram to query for interesting samples, then the selected tail and the opposite end will be highlighted in different colors. The scatter plot will be replaced by a collection of scatter plots of selected samples, which allow users to check details on each selected samples. After brushing, a new section of table will show up. From the table, Users can also access the test statistics of each term for selected samples.

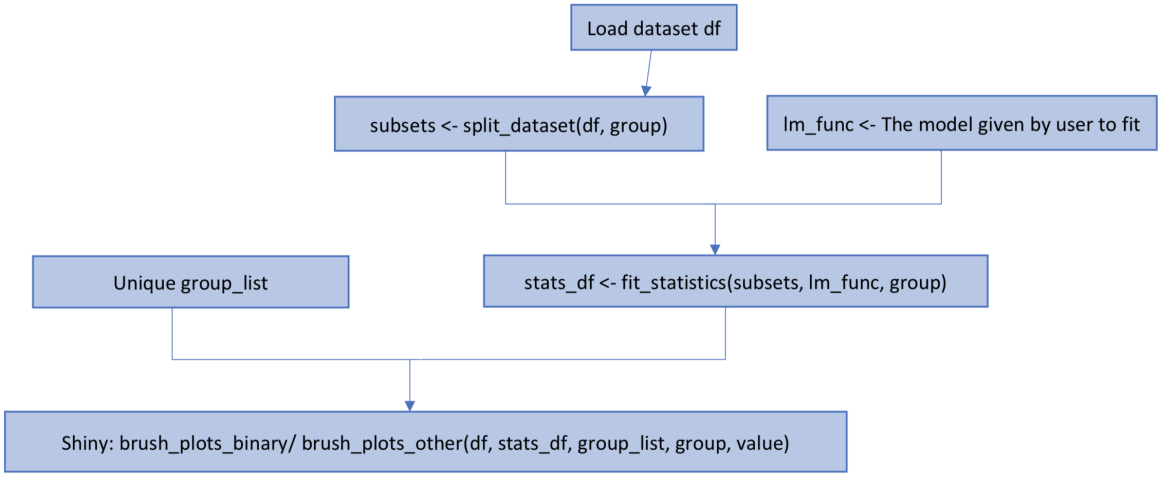

Figure 1: Workflow

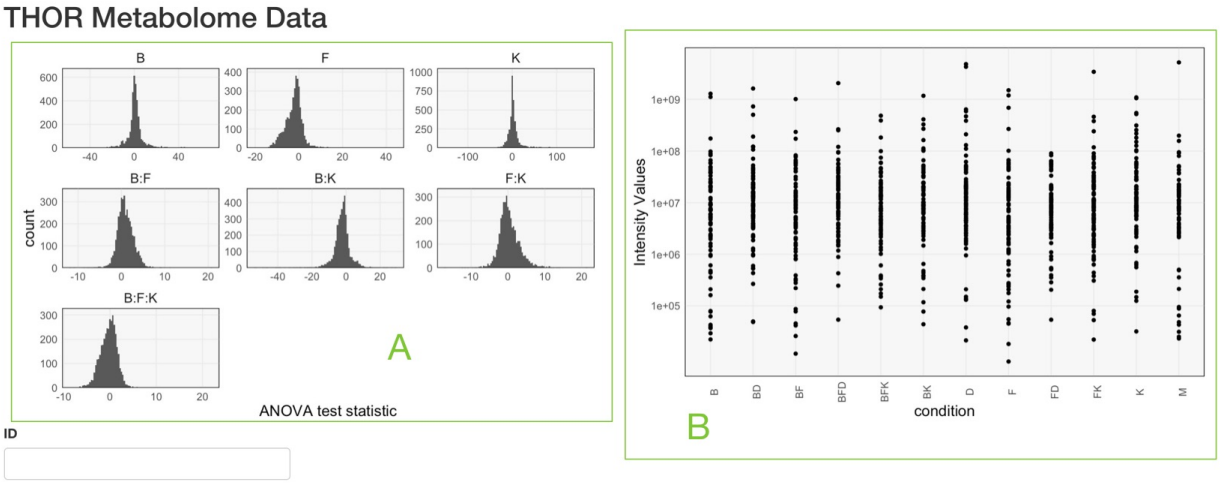

Figure 2: Interface before interaction

A: Histograms of test statistics, brush through any one of the histogram to select samples;
B: Before brushing, a scatter plot of all samples will appear to give an overall distribution of the dataset.

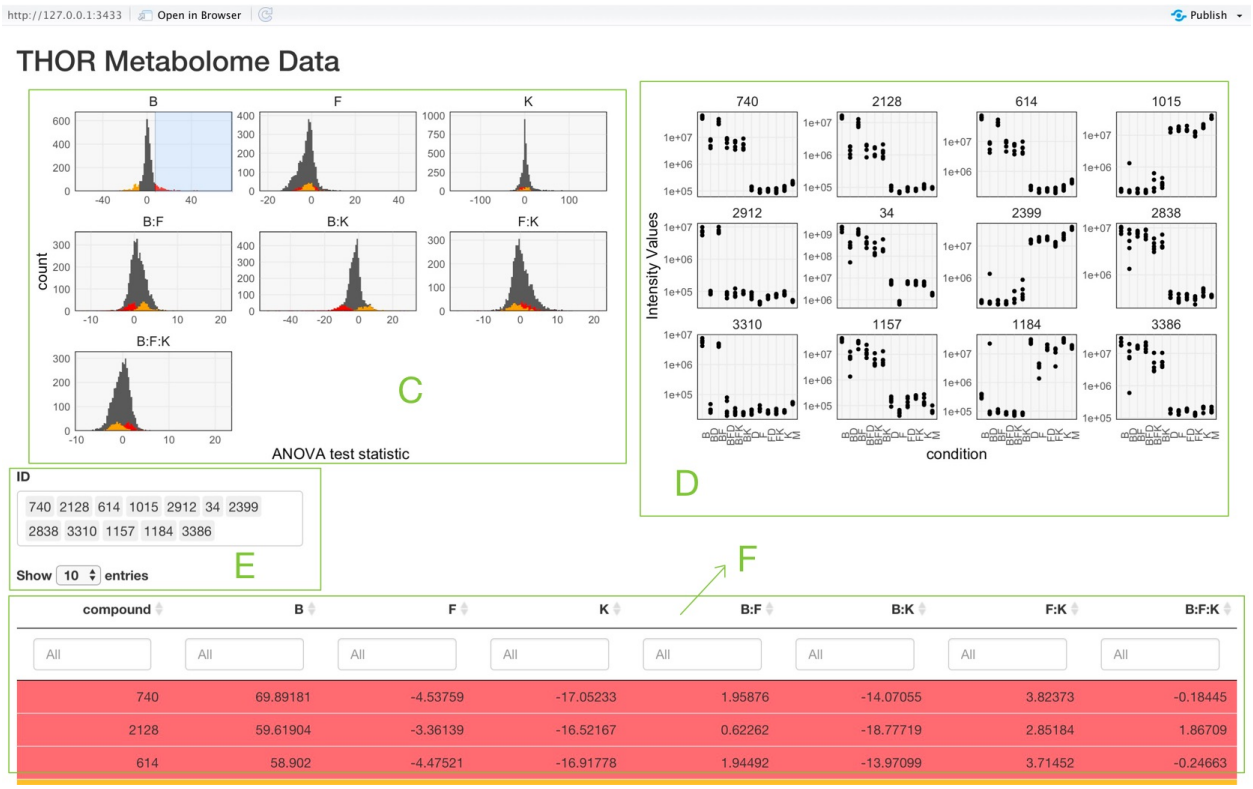

Figure 3: Interface after brushing

C: After brushing, test statistics greater than and less than the threshold will be highlighted (red for positive and orange for negative values). The selected samples will be highlighted in all histograms;
D: The scatter plots of top 12 samples in the table of selected samples will be shown here. By adding or deleting the chosen ID on the left select menu, users can choose to see the scatter plot of the sample they have interests on;
E: By default, only 12 samples will be chosen at first. Users can use the drop down menu to follow the samples of interests;
F: The table of selected samples in the brushed area. Color corresponds to the color in the upper histograms.

Cases Study

THOR Example

The THOR dataset contains 3882 compounds with five replicate measurements under 12 different conditions. From the data, we want to find out the community effect of the Thor system.

This plot shows the overall condition of the data points before any interaction. From the scatter plot, we can notice that the distribution of the data points are relatively uniform and there is no extreme outliers.

In the figure below, we brushed compounds with large, positive F effects. The reflected negative compounds are in orange. The associated FK effects are mostly positive and the associated BFK effects are mostly negative (compare the orange bulks with 0). A possible explanation is that these orange compounds are consumed by F, which is why they decrease in the presence of F. The positive FK effects means those compounds are not being consumed so much when K is present. But when all three are present, the compound is less abundant again, it is an evidence for B protecting F.

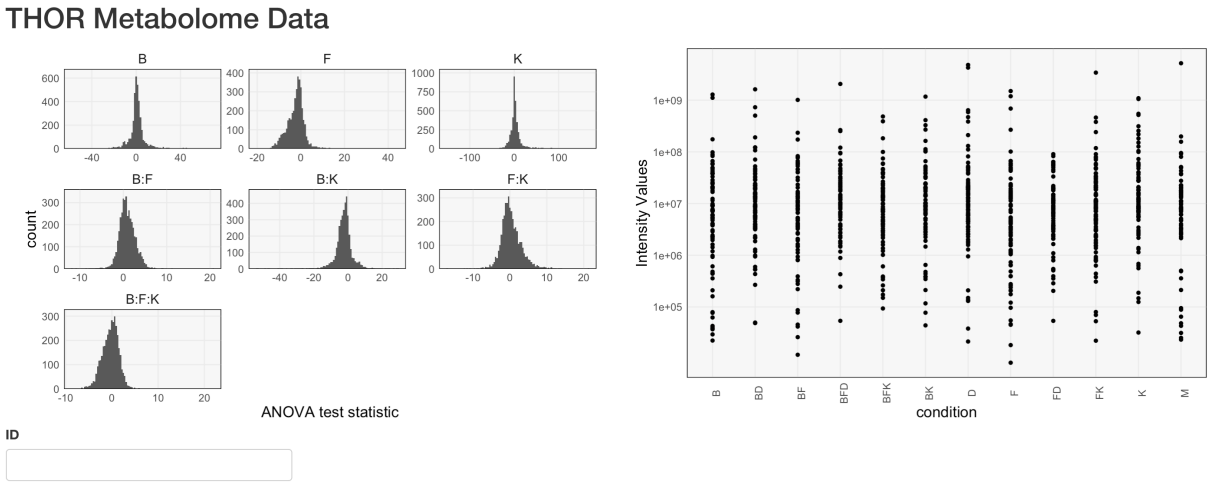

Figure 4: Example1.1

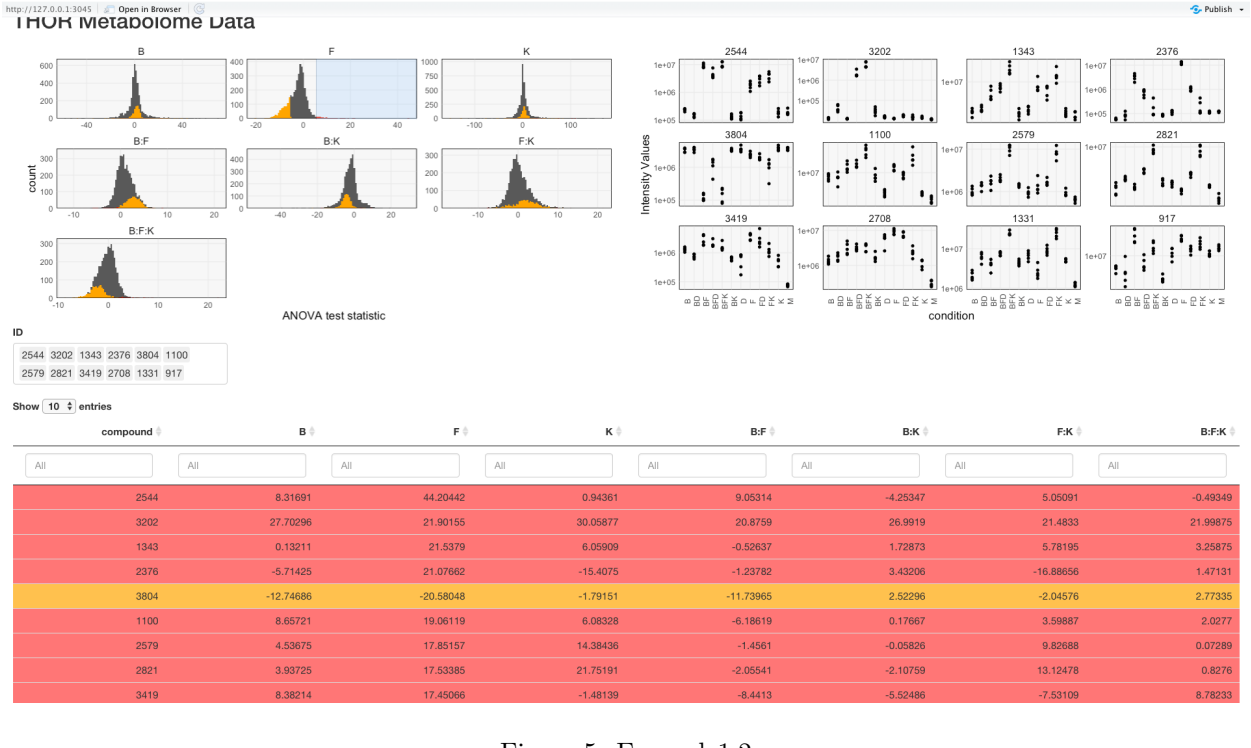

Figure 5: Example1.2

House Price Example

The House price dataset contains the information of 20640 areas around some longitude and latitude and including characteristics like population, households, and median income in those areas. From the data, we want to find out some trends among the house price and the condition of a house.

Since the distribution of houses is not uniform among all areas, some area may contain hundreds of houses but others may only have a few houses. For areas with limited number of houses, it is not representative to fit a model and compare with other ares. Therefore, to solve this problem, in the data process phase we group the houses into 2064 clusters based on their longitude and latitude using k-mean clustering so that the number of houses in each cluster will be large enough to study.

From the overall distribution of the dataset, we can notice that the houses will top high price are normally in ares with small population and the house price of areas with heavy population is relatively low.

In the following figure, we brushed houses with positive households parameters. While, the corresponding households and median income interaction parameters are negative, which represents that median income restrains the positive effects of households. When population is also added, the tree-way interaction term seems to have a positive effects again.

For house price dataset, it is normal to fit a mode and use the model to make predictions for house price. While by performing multiple hypothesis testing, it is convenient to compare the trend within different areas and the scatter plots allow us to notice the detail of each invidual house in the area.

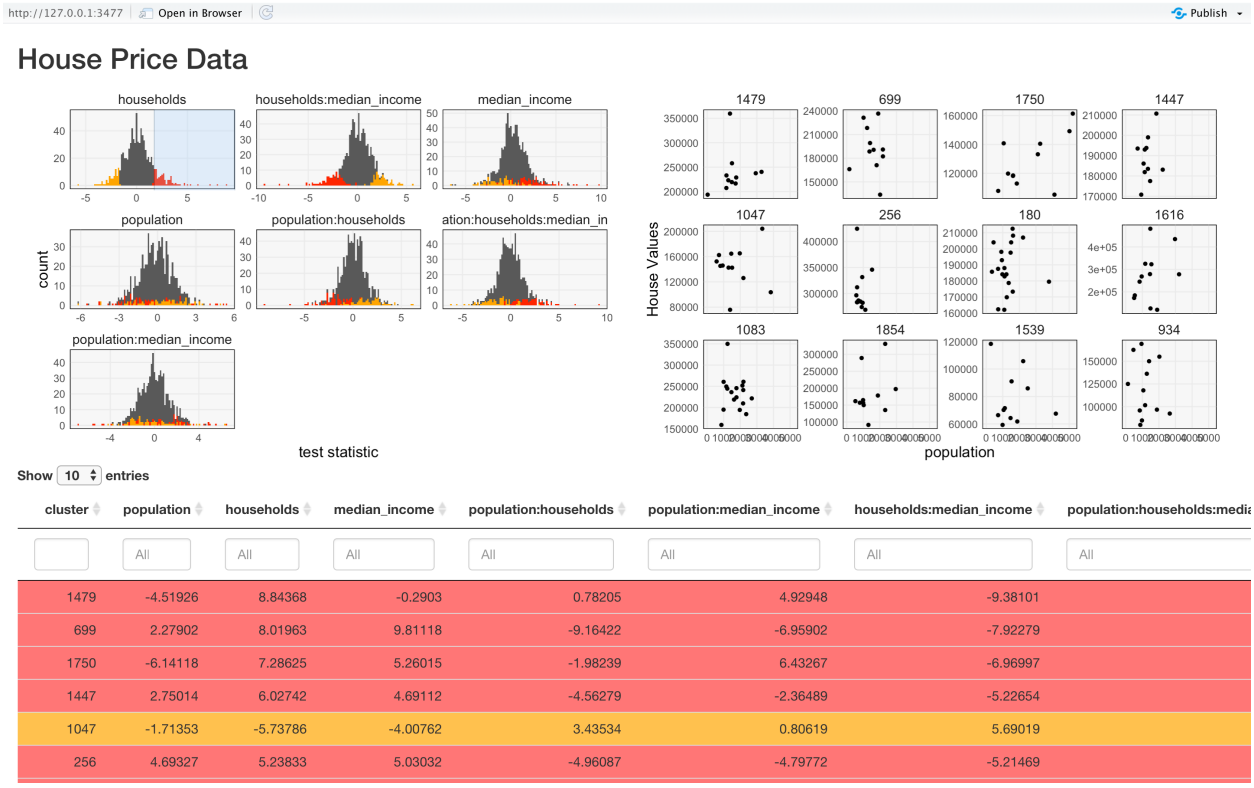

Figure 6: Example2

Discussion

This package help to perform a multiple hypothesis testing for similar samples in a large number of groups. After clean the data into the type of input data on the pipeline, given a model, we can easily apply the model

to all the groups and visualize test statistics. Then by brushing through linked diagrams, we can interpret the data in a more intuitive and efficient way.

However, this package has limited ability to check the correctness and effectiveness of the model and we do not implement the multiple hypothesis testing outputs, so it is a more conceptual link. We give users the freedom to use their own model and believe that it is valid. In the further study, we could have a more complete process of hypothesis testing including null and alternative hypothesis. We should also pay attention to the p-value and think about how to avoid rejecting the null hypothesis due to the tolerance of significant level when dealing with a bunch of test statistics. Also, there is only one brush per panel so the information get from the plots are limited. We are not allowed to select several regions form the same panel and compare them. Lastly, the data type of variables is limited either binary or non-binary type. No better strategies for datasets with mixing of binary and non-binary variables.